afi

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/070,092	06/28/2002	Gilbert Wolrich	10559-309US1	7323

7590 11/02/2005

Fish & Richardson
225 Franklin Street
Boston, MA 02110-2804

EXAMINER	
RIZZUTO, KEVIN P	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 11/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	10/070,092	WOLRICH ET AL.	
	Examiner	Art Unit	
	Kevin P Rizzuto	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 9/9/05
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-27 have been examined.

2. Acknowledgement of papers filed: RCE filed on 9/9/05. The papers filed have been placed on record.

Withdrawn Claim Objections/Rejections

3. Applicant, via amendment, has overcome the objections to the claims 4, 6, 9, 11-27 set forth in the previous Office Action. Consequently, the examiner has withdrawn these objections.

4. Applicant, via amendment, has overcome the objection to the title set forth in the previous Office Action. Consequently, the examiner has withdrawn this objection.

5. Applicant, via amendment, has overcome the 35 U.S.C. 112 1st paragraph Rejection to claims 3/8, 12/17 and 21-26 set forth in the previous Office Action. Consequently, the examiner has withdrawn these Rejections.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-8, 10-17 and 19-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao, U.S. Patent 4,569,016, in view of Hennessy and Patterson, Computer Organization and Design and further in view of Hennessy and Patterson, Computer Architecture, A Quantitative Approach.

8. As per claim 1, Hao teaches a hardware-based multithreaded processor comprising a plurality of microengines, each of the microengines comprising:

-A control store: (I-cache 18, figure 1)

-Controller logic: (The IAR 34 and IAR+4 36 supply the address for fetching instructions as shown in figures 2A and 2B. There is inherently controller logic within the "IAR" 34 and/or "IAR+4" 35 or within a separate not-shown controller, which controls what instruction to fetch from the I-Cache, i.e., the next sequential or a branched-to instruction.) (Column 24, lines 4-23)

-And an execution box data path including an arithmetic logic unit (ALU) (Mask & Rotate Logic 56, figure 2B) and a general purpose register set (general purpose registers 30, figure 2A), the ALU performing functions in response to instructions: (Column 6, lines 31 to column 7, line 19)

-One of the instructions causing the ALU to selectively load any specified combination of bytes of data within a transfer register associated with one microengine with a shifted value of an operand that preserves the bytes of data that are not loaded (Figure 3, Column 26, lines 36-46, Column 12 line 55 to column 13, line 15 and Column 21, lines 17-25; The registers used by the instructions have data transferred in to them and out of them, therefore they are transfer registers. Any time an instruction specifies

bytes to be selectively loaded, they are in fact selectively loaded, therefore, Hao teaches "one of the instructions causing the ALU to selectively load any specified combination of bytes". Furthermore, "any" is defined as, " One, some, every, or all without specification." (American Heritage Dictionary, 4th Ed.) Therefore, since Hao teaches selectively loading more than one combination of bytes (dependent on the mask), Hao teaches "the ALU to selectively load any specified combination of bytes".)

9. While Hao does teach a transfer register associated with one microengine, Hao fails to teach that the microengine is one of a plurality of microengines or that the microengines contain context event switching logic.

10. Hennessy and Patterson teach that it is beneficial to connect multiple microengines together to become a multiprocessor (page 712 of Computer Organization and Design). Duplicating the microengine taught by Hao and using them together as a multiprocessor system as taught in Hennessy and Patterson would cause the microengine of Hao to be one of a plurality of microengines. The multiprocessor system provides increased processing speeds and capabilities, which would have provided the motivation to one of ordinary skill in the art to combine the teachings of Hao with Hennessy and Patterson (Page 712).

11. Hao, in view of Hennessy and Patterson, Computer Organization and Design, fails to teach context switching logic.

12. Hennessy and Patterson, A Quantitative Approach, teach that context switching logic allows a processor to run multiple processes at once (i.e., time sharing). Time-

sharing the processor among multiple users/processes causes a single processor to create the illusion that all users have their own machine. (Pages 447-449)

13. It would have been obvious to one of ordinary skill in the art to add context switching logic to the microengines taught in Hao, in view of Hennessy and Patterson, Computer Organization and Design, as taught in Hennessy and Patterson, A Quantitative Approach. Causing a single processor to create the illusion of multiple processors, which allows multiple users to simultaneously each have their own process running on the single processor, would have provided the motivation to one of ordinary skill.

14. As per claim 19, Hao teaches a computer instruction comprising:

-An instruction that loads one or more bytes of data within a transfer register associated with one microengine with a shifted value of an operand that preserves the bytes of data that are not loaded (Figure 3, Column 26, lines 36-46, Column 12 line 55 to column 13, line 15 and Column 21, lines 17-25; The registers used by the instructions have data transferred in to them and out of them, therefore they are transfer registers. Furthermore, "any" is defined as, " One, some, every, or all without specification." (American Heritage Dictionary, 4th Ed.) Therefore, since Hao teaches selectively loading more than one combination of bytes (dependent on the mask), Hao teaches "the ALU to selectively load any specified combination of bytes".))

15. While Hao does teach a transfer register associated with one microengine, Hao fails to teach that the microengine is one of a plurality of microengines.

16. Hennessy and Patterson teach that it is beneficial to connect multiple microengines together to become a multiprocessor (page 712). Duplicating the microengine taught by Hao and using them together as a multiprocessor system as taught in Hennessy and Patterson would cause the microengine of Hao to be one of a plurality of microengines. The multiprocessor system provides increased processing speeds and capabilities, which would have provided the motivation to one of ordinary skill in the art to combine the teachings of Hao with Hennessy and Patterson (Page 712).

17. As per claims 2, 11 and 20, the computer instruction/method of claims 1, 10, and 19, further comprising:

-A bit mask that specifies which of the bytes of data are affected. (Instructions in M-form have a bit Mask encoded within bits 21-31 and other instructions generate a bit-mask (Columns 11-12, Tables 2(a) and 2(b), Column 13, the descriptions of instructions RIMI and RIMN and Column 14, lines 18-25 and descriptions of the X-form instructions)

18. As per claims 3, 12 and 21, wherein the instruction further comprises a field that indicates a left shift n bits, where n is a number from one to thirty-one (RIMI and RIMN encode 5 shift bits that indicate a left shift of 0 to thirty-one bits. (Column 12 line 42 to Column 13, line 15) Tables 2(a) and 2(b) in columns 11 and 12 show the SH field being 5 bits. A rotate left operation shifts bits to the left and therefore is a left shift operation. (Column 12, lines 61-63) Hao also teaches strictly left and right shift instructions without the rotate function. (Column 14, lines 18-24))

19. As per claims 4, 13 and 22, wherein the instruction further comprises a field that indicates a left shift by an amount specified in five bits of the first operand of a previous instruction, where the lower five bits is a number from one to thirty-one. (For the RMI and RNM instructions, the shift amount is indicated by an amount specified in 5 bits (bits 27-31) of the RB register. It is inherent that in order for data to be in the RB register to specify a shift amount, it must have been an operand of a previous instruction (Column 13, lines 29-42 and lines 55-70 and Column 25, lines 56-66))

20. While Hao does teach that the shift amount is specified in 5 bits of a first operand of a previous instruction, Hao does not teach that the 5 bits are in a lower five bits of the operand.

21. It would have been obvious to one of ordinary skill in the art at the time the invention was made to place the 5 bits that specify the shift amount into the lower 5 bits of register RB instead of the upper 5 bits since it has been held that a mere rearrangement of parts that does not modify the operation of the device does not make said device patentable. (*In re Japikse,* 181 F.2d 1019, 86 USPQ 70 (CCPA 1950)).

22. As per claims 5, 14 and 23, wherein the instruction further comprises a field that indicates a right shift n bits, where n is a number from one to thirty-one. (RIMI and RIMN encode 5 shift bits that indicate a left shift of 0 to thirty-one bits. Column 12 line 42 to Column 13, line 15. Tables 2(a) and 2(b) in columns 11 and 12 show the SH field being 5 bits. A rotate left operation shifts bits to the left, however the rotate left instructions allow rotate right instructions to be performed by a rotate left of 32-N, where N is the number of positions to rotate right. A rotate right operation includes shifting bits to the

right and therefore is a shift right operation. (Column 13, lines 12-15 and Column 12, lines 61-63) Hao also teaches strictly left and right shift instructions without the rotate function. (Column 14, lines 18-24))

23. As per claims 6, 15 and 24, wherein the instruction further comprises a field that indicates a right shift by an amount specified in a lower five bits of the first operand of a previous instruction, where the lower five bits is a number from one to thirty-one. (For the RMI and RNM instructions, the shift amount is indicated by an amount specified in 5 bits (bits 27-31) of the RB register. (Column 13, lines 29-42 and lines 55-70 and Column 25, lines 56-66)) A rotate left operation shifts bits to the left, however the rotate left instructions allow rotate right instructions to be performed by a rotate left of 32-N, where N is the number of positions to rotate right. A rotate right operation includes shifting bits to the right and therefore is a shift right operation. (Column 13, lines 12-15 and Column 12, lines 61-63) It is inherent that in order for data to be in the RB register to specify a shift amount, it must have been an operand of a previous instruction.

24. While Hao does teach that the shift amount is specified in 5 bits of a first operand of a previous instruction, Hao does not teach that the 5 bits are in a lower five bits of the operand.

25. It would have been obvious to one of ordinary skill in the art at the time the invention was made to place the 5 bits that specify the shift amount into the lower 5 bits of register RB instead of the upper 5 bits since it has been held that a mere rearrangement of parts that does not modify the operation of the device does not make said device patentable. (*In re Japikse,* 181 F.2d 1019, 86 USPQ 70 (CCPA 1950)).

26. As per claims 7, 16 and 25 wherein the instruction further comprises a field that indicates a left rotate n bits, where n is a number from one to thirty-one. (RIMI and RIMN encode 5 rotate bits that indicate a left rotate of 0 to thirty-one bits. Column 12 line 42 to Column 13, line 15. Tables 2(a) and 2(b) in columns 11 and 12 show the SH field being 5 bits.

27. As per claims 8, 17 and 26, wherein the instruction further comprises a field that indicates a right shift n bits, where n is a number from one to thirty-one. (RIMI and RIMN encode 5 shift bits that indicate a left shift of 0 to thirty-one bits. Column 12 line 42 to Column 13, line 15. Tables 2(a) and 2(b) in columns 11 and 12 show the SH field being 5 bits. A rotate left operation shifts bits to the left, however the rotate left instructions allow rotate right instructions to be performed by a rotate left of 32-N, where N is the number of positions to rotate right. A rotate right operation includes shifting bits to the right and therefore is a shift right operation. (Column 13, lines 12-15 and Column 12, lines 61-63) Hao also teaches strictly left and right shift instructions without the rotate function. (Column 14, lines 18-24))

28. As per claim 10, Hao teaches a method of operating a processor comprising:

-Loading one or more bytes of data within a register associated with one microengine with a shifted value of an operand and clearing the bytes of data that are not loaded: (Figure 3, Column 26, lines 36-46, Column 13, lines 1-15 and lines 43-54, Column 14, lines 15-25, and column 20, line 58 to column 21, line 17). The registers used by the instructions have data transferred in to them and out of them, therefore they are transfer registers.)

29. While Hao does teach a transfer register associated with one microengine, Hao fails to teach that the microengine is one of a plurality of microengines.

30. Hennessy and Patterson teach that it is beneficial to connect multiple microengines together to become a multiprocessor (page 712). Duplicating the microengine taught by Hao and using them together as a multiprocessor system as taught in Hennessy and Patterson would cause the microengine of Hao to be one of a plurality of microengines. The multiprocessor system provides increased processing speeds and capabilities, which would have provided the motivation to one of ordinary skill in the art to combine the teachings of Hao with Hennessy and Patterson (Page 712).

31. Claims 9, 18 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao, U.S. Patent 4,569,016 in view of Hennessy and Patterson, Computer Organization and Design, and further in view of Kiuchi, U.S. Patent 5,832,258.

32. Hao, in view of Hennessy and Patterson, teach the computer instruction or method of claims 1, 10 and 19. Hao further teaches that a condition register (Column 9, tables 1(a) and 1(b)) is updated for the different shift/rotate, mask and merge instructions (Column 10, lines 53-55). The condition register contains ALU condition codes that are set based on results of executed instructions, and it is updated differently, depending on the instruction being executed (Columns 13 and 14, the individual descriptions of instructions)

33. However, an optional token that is set by a programmer and specifies to set arithmetic logic unit (ALU) condition codes based on the result is not taught.

34. Kiuchi teaches the optional updating of a condition register depending on an optional token (CC field) encoded in an instruction. When the CC field is set to 0001, an unconditional instruction does not update the condition codes in a condition register (Column 37, lines 59-62). The Condition Code decoder is explained in Columns 31 and 32, the CC codes and their meanings are taught in Table 1, Columns 37 and 38. This provides the benefit of greater program flexibility and a reduction in code size ("Objects of the invention", Column 2 and "Advantages of Conditional Data Operation with No Condition Code Update," Columns 51 and 52). It would have been obvious to one of ordinary skill in the art to combine the invention of Kiuchi with the invention of Hao in view of Hennessy and Patterson because of the benefits Kiuchi teaches.

Response to Arguments

35. Applicants arguments filed on 1/24/2005 have been fully considered but they are not persuasive.

36. Applicant argues the novelty/rejection of claims 1, 10 and 19 by stating:

> a. "**<u>Given the very specific format that the resultant constructed mask in Hao may assume, Hao's mask cannot specify any combination of bits corresponding to the positions in the destination register that should be loaded with the source operand's value.</u>** For example, Hao's mask cannot load the source operand into bytes 1 and 3 of the destination register since to so, the mask would have to be constructed as: "11111111000000001111111100000000". Such a mask cannot be constructed using the mask field in any of Hao's instructions since it includes two strings of consecutive 1's and two strings of consecutive 0's, which Hao's instruction mask field is incapable of producing.
>
> Since Hao's mask cannot specify any combination of byte positions in a destination register, Hao therefore does not disclose or suggest "causing the ALU selectively load any specified combination of bytes of data within a transfer register associated with one of the plurality of microengines with a shifted value of an operand that preserves the bytes of data that are not loaded", as required by applicant's independent claim 1." (Emphasis added by Examiner)

37. These arguments are not found persuasive for the following reasons:

b. Contrary to what Applicant argues, Hao teaches a mask that can specify multiple different combinations for selectively loading data into a specified register. The American Heritage Dictionary, 4th Ed., defines any as, "One, some, every, or all without specification." (American Heritage Dictionary, 4th Ed.) Since Hao teaches multiple combinations can be specified, Hao's mask qualifies as being capable of specifying "any combination" (Specifically, from the definition of any, multiple combinations falls under "one or some"). Furthermore, Applicants argument states, "Hao's mask cannot specify any combination of bits," which is clearly not true, since Applicant admits that Hao can specify combinations of bits in the paragraph directly above the argument on page 10 of the Applicant's "Remarks".

c. Applicant submits an example following the argument, wherein Applicant's invention can selectively load bytes 1 and 3, whereas Hao cannot. However, this limitation is not required by the present claim language.

38. Applicant argues the novelty/rejection of claims 9, 18 and 27 by stating:

d. "The examiner, however, argued that Kiuchi describes this feature. Applicant respectfully disagrees. Kiuchi describes a digital signal processor (Abstract). While Kiuchi discloses an instruction word that also includes a condition code field that identifies a predefined condition and also identifies whether the condition code register should be updated when the data processing operation is performed by the execution unit (abstract), nowhere does Kiuchi disclose or suggest that such a condition code specifies loading ALU condition codes, and certainly it does not disclose or suggest doing so based on a result formed in the ALU, as required by applicant's amended claim 9.

Since none of the reference[s] cited by the examiner disclose[s] or suggest[s], alone or in combination, the feature of "an optional token that is set by a programmer and

specifies to set arithmetic logic unit (ALU) condition codes based on a result formed in the ALU," applicant's claim 9 is thus patentable over the cited art."

39. These arguments are not found persuasive for the following reasons:

e. Applicant's arguments have failed to address the cited portions of Kiuchi, and has merely cited the abstract of Kiuchi while arguing what is not taught by Kiuchi. Applicant's attention is directed towards the disclosure of Kiuchi, col. 37, line 10 to col. 38, line 67. Kiuchi describes how if the CC field is set appropriately, the CR register will be updated with condition codes based on a result from the ALU (Specifically, col. 37, lines 10-37 describes the updating). Kiuchi specifically states, "***As shown in FIG. 10, the C, N, Z, V, VR or GC flags may be generated by either the ALU 1712*** or the BPU 1714." However, if the CC field is set to "0001", the condition codes in the CR register are not updated. Therefore, looking at the cited portions of Kiuchi clearly teaches the limitation of "an optional token that is set by a programmer and specifies to set arithmetic logic unit (ALU) condition codes based on a result formed in the ALU.

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the

objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin P Rizzuto whose telephone number is (571) 272-4174. The examiner can normally be reached on M-F, 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4174. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KPR

Eddie Chan

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100